WHAT IS CLAIMED IS:

1. A method for setting firewall policy for an application, comprising:

receiving a first parameter comprising information about an application;

receiving a second parameter comprising information about a user of the application;

accessing security level information relating to the first and second parameters; and

setting a firewall policy for the application and the user utilizing the security level information.

2. The method of claim 1, wherein accessing security level information comprises calling a GetRules method to return rule templates available for the application and user.

3. The method of claim 1, further comprising, receiving a third parameter regarding trusted contacts, and wherein setting a firewall policy comprises setting the firewall policy for the application and the user and the trusted contacts utilizing the security level information.

4. The method of claim 1, wherein setting the firewall policy comprises:

setting a default setting for the application; and

selecting the default setting for the user.

5. A computer-readable medium having stored thereon a data structure, the data structure comprising;

a first data field representing an application;

a second data field representing a user of the application; and

a third data field representing available security settings for the user utilizing the application.

6. The computer-readable medium of claim 5, wherein the data structure comprises a data object.

7. An object model for managing a service on a computer, the object model comprising:

a policy object model used to specify one or more policies that the service supports; and

a policy engine platform for interacting with said one or more policies for the service and at least one component that

actually performs the service, and to provide said one or more policies to said at least one component.

8. The object model of claim 7, wherein the policy engine platform comprises a rule editor for adding an additional policy in accordance with the policy object model.

9. The object model of claim 8, wherein the rule editor is also configured to delete a policy.

10. The object model of claim 8, wherein the rule editor is also configured to edit a policy.

11. The object model of claim 7, wherein the policy engine platform comprises a setting editor configured to automatically generate a policy based upon an application and user combination.

12. The object model of claim 11, wherein the setting editor generates a plurality of policies, and is further configured to permit a user to select from the plurality.

13. The object model of claim 12, wherein the setting editor is further configured to permit setting one of the plurality as a default policy.

14. The object model of claim 7, wherein the policy engine platform comprises a rule explorer for providing a view of the one or more policies.

15. The object model of claim 7, wherein the policy object model comprises a policyrule object usable to generate policy, the policyrule object comprising a condition property and an action property, wherein a policy generated by the policyrule object is configured to perform an action in the action property responsive to a condition in the condition property being met.

16. The object model of claim 7, wherein the service is a firewall service.

17. The object model of claim 7, wherein the policy engine platform is configured to deny providing said one or more policies to the component if a requestor is not authorized.

18. The object model of claim 17, wherein determining whether a requestor is authorized comprises comparing a provider rank for the requestor against a permitted rank, and if the provider rank for the requestor does not meet or exceed the permitted rank, denying the requestor.

19. A method of managing a service on a computer, the method comprising:

specifying, via a policy object model, one or more policies that the service supports; and

interacting, via a policy engine platform, with said one or more policies for the service and at least one component that actually performs the service; and

providing, via the policy engine platform, said one or more policies to said at least one component.

20. The method of claim 19, further comprising automatically generating a policy based upon an application and user combination.

21. The method of claim 20, further comprising generates a plurality of policies, and permitting a user to select from the plurality.

22. The method of claim 21, further comprising setting one of the plurality as a default policy.

23. The method of claim 22, further comprising authorizing a user prior to providing.

24. An object model for managing a firewall service on a computer, the object model comprising a policy object model used to specify one or more policies that the firewall service supports, the policy model comprising a policyrule object usable to generate policy, the policyrule object comprising a condition property and an action property, wherein a policy generated by the policyrule object is configured to perform an action in the action property responsive to a condition in the condition property being met.

25. The object model of claim 24, further comprising an IPSecRule derived from the policyrule object, the IPSecRule being configured to trigger an IPSec callout when an IPSec

condition is matched, and to indicate configuration parameters for securing traffic related to the callout.

26. The object model of claim 25, wherein the IPSecRule evaluates a standard 5-tuple to determine if a condition has been met.

27. The object model of claim 24, further comprising a KeyingModuleRule derived from the policyrule object, the KeyingModuleRule being configured to select which key negotiation module to use when there is no existing secure channel to a remote peer.

28. The object model of claim 27, wherein the KeyingModuleRule evaluates a standard 5-tuple to determine if a condition has been met.

29. The object model of claim 24, further comprising a IKERule derived from the policyrule object and configured to specify the parameters for carrying out Internet Key Exchange key negotiation protocol.

30. The object model of claim 29, wherein the IKERule evaluates a local address and a remote address to determine if a condition has been met.

31. The object model of claim 29, wherein the IKERule comprises an IKEAction action property that defines the authentication methods for performing Internet Key Exchange key negotiation protocol.